

SQL Programming (3)

- ◆ SET Operators
 - ◆ UNION, INTERSECT, MINUS
- ◆ DML
 - ◆ INSERT, UPDATE
 - ◆ DELETE, DROP
- ◆ View
- ◆ NULL

Readings: Sections 6.2.5, 6.5, 8.1 of Textbook.

The Village Cinema Database Instance

Movie

mvID	Title	Rating	Rel_date	Length	Studio
1	Angels & Demons	M	14-05-2009	138	Sony Pictures
2	Coco Avant Chanel	PG	25-06-2009	108	Roadshow
3	Harry Potter and the Half-Blood Prince	M	15-07-2009	153	Roadshow
4	The Proposal	PG	18-06-2009	107	Disney
5	Ice Age: Dawn of the Dinosaurs	PG	01-07-2009	94	20th Century Fox

Classification

mvID	Genre
1	Drama
2	Drama
3	Drama
3	Action
3	Adventure
4	Comedy
5	Comedy
5	Animated

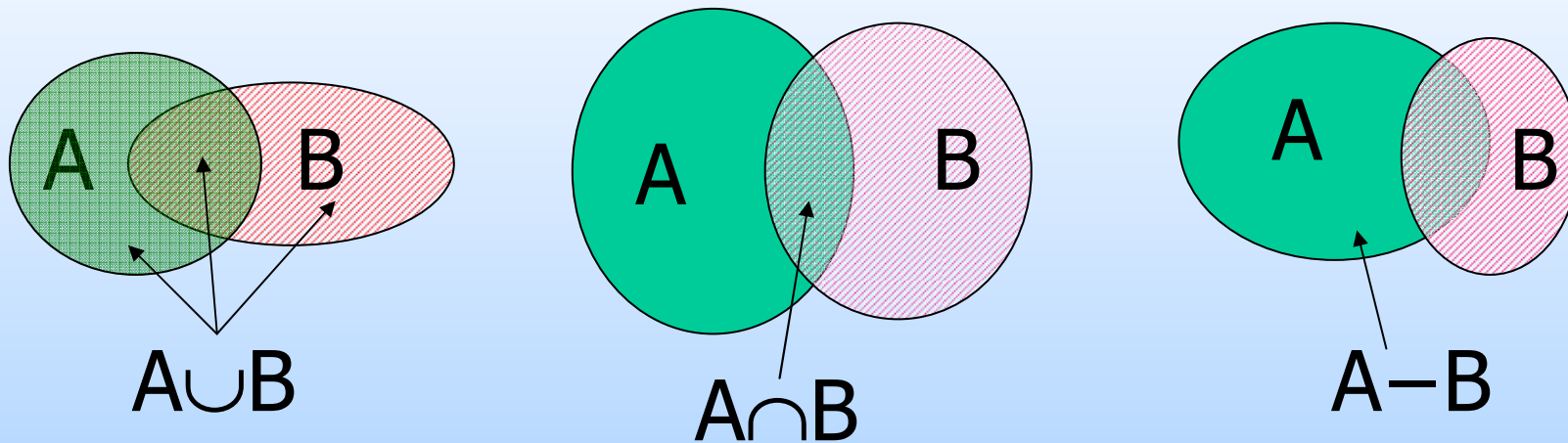
Cast

mvID	Actor
1	Tom Hanks
2	Audrey Tautou
2	Benolt Poelvoorde
2	Alessandro Nivola
2	Marie Gillain
3	Daniel Radcliffe
3	Emma Watson
3	Rupert Grint
4	Sandra Bullock
4	Ryan Reynolds
4	Malin Akerman
4	Mary Steenburgen
4	Betty White
5	John Leguizamo
5	Queen Latifah
5	Denis Leary
5	Ray Ramono
5	Chris Wedge

Direct

mvID	Director
1	Ron Howard
2	Anne Fontaine
3	David Yates
4	Anne Fletcher
5	Carlos Saldanha
5	Mike Thurmeier

Set Operations: Union, Intersection and Difference



Set operations:

Union, Intersection, and Difference

- ◆ Union, intersection, and difference of relations are expressed by the following expressions involving subqueries:
 - ◆ (<subquery>) UNION (<subquery>)
 - ◆ (<subquery>) INTERSECT (<subquery>)
 - ◆ (<subquery>) MINUS (<subquery>)

Union

- ◆ Find the movies (mvID) that have either “Tom Hanks” or “Audrey Tautou”.
- ◆ Solution:
 1. Find the movies by “Tom Hanks”.
 2. Find the movies by “Audrey Tautou”.
 3. Union the results from 1 and 2, and form the solution.

Solution

```
(select mvid
from Cast
where actor='Tom Hanks')
UNION
(select mvid
from Cast
where actor='Audrey Tautou');
```

MVID
1

MVID
2

MVID
1
2

Intersection

- ◆ Find the movies (mvID) that have both “Marie Gillain” and “Audrey Tautou”.
- ◆ Solution:
 1. Find the movies that have “Marie Gillain”.
 2. Find the movies that have “Audrey Tautou”.
 3. Find the intersection of results from 1 and 2, which is the solution.

Solution

Movies that have
"Marie Gillain".

```
(Select mvID  
From Cast  
Where actor='Marie Gillain')
```

INTERSECT

```
(Select mvID  
From Cast  
Where actor='Audrey Tautou');
```

Movies that have
"Audrey Tautou".

Brackets "(...)"
are necessary.

Difference (MINUS)

- ◆ Find the movies that are *only* in the genre "Drama".
 1. Find the movies in the genre "Drama".
 2. Find the movies in genres other than "Drama".
 3. Take the difference of 1 and 2, and form the solution.

Solution

```
(select mvID
from Classification
where genre='Drama')
minus
(select mvID
from Classification
where genre != 'Drama');
```

MVID
1
2

MVID
1
2
3

MVID
3
3
4
5
5

Set operators Remove duplicates!

- ◆ The default SELECT-FROM-WHERE statement keeps duplicates (the bag semantics).
- ◆ The default union, intersection, and difference expressions remove duplicates (set semantics).

Set operators remove duplicates ...

- ◆ Find the movies (mvID) that are not in the genre "Drama"

```
select mvID  
from Classification  
where genre != 'Drama';
```

MVID
3
3
4
5
5

```
select DISTINCT mvID  
from Classification  
where genre != 'Drama';
```

MVID
3
4
5

Set operators remove duplicates ...

```
(select mvid  
from Classification  
where genre != 'Drama')
```

UNION

```
(select mvid  
from Classification  
where genre != 'Drama');
```

MVID
3
3
4
5
5

MVID
3
3
4
5
5

MVID
3
4
5

Data Manipulation Using SQL

◆ Insert a row into a table

- ◆ INSERT INTO ... VALUES (...)

◆ Update rows in a table

- ◆ UPDATE *table* SET column=value WHERE ...

◆ DELETE rows in a table

- ◆ DELETE FROM *table* WHERE

The Village Cinema Database Instance again

Movie

mvID	Title	Rating	Rel_date	Length	Studio
1	Angels & Demons	M	14-05-2009	138	Sony Pictures
2	Coco Avant Chanel	PG	25-06-2009	108	Roadshow
3	Harry Potter and the Half-Blood Prince	M	15-07-2009	153	Roadshow
4	The Proposal	PG	18-06-2009	107	Disney
5	Ice Age: Dawn of the Dinosaurs	PG	01-07-2009	94	20th Century Fox

Classification

mvID	Genre
1	Drama
2	Drama
3	Drama
3	Action
3	Adventure
4	Comedy
5	Comedy
5	Animated

Cast

mvID	Actor
1	Tom Hanks
2	Audrey Tautou
2	Benolt Poelvoorde
2	Alessandro Nivola
2	Marie Gillain
3	Daniel Radcliffe
3	Emma Watson
3	Rupert Grint
4	Sandra Bullock
4	Ryan Reynolds
4	Malin Akerman
4	Mary Steenburgen
4	Betty White
5	John Leguizamo
5	Queen Latifah
5	Denis Leary
5	Ray Ramono
5	Chris Wedge

Direct

mvID	Director
1	Ron Howard
2	Anne Fontaine
3	David Yates
4	Anne Fletcher
5	Carlos Saldanha
5	Mike Thurmeier

Updates to the database

- ◆ Insert a new movie with mvID=6:
 - ◆ Title: The Da Vinci Code.
 - ◆ Rating: M.
 - ◆ Release date: 18-May-2006.
 - ◆ Length: Unknown.
 - ◆ Production studio: Unknown.
 - ◆ Genre: Mystery, Thriller.
 - ◆ Cast: Tom Hanks, Audrey Tautou.
 - ◆ Director: Ron Howard.

INSERT

```
INSERT INTO Movie  
VALUES(6, 'The Da Vinci Code', 'M', '18-05-2006', NULL, NULL);
```

```
INSERT INTO Classification  
VALUES(6, 'Mystery');
```

```
INSERT INTO Classification  
VALUES(6, 'Action');
```

```
INSERT INTO Cast  
VALUES(6, 'Tom Hanks');
```

```
INSERT INTO Cast  
VALUES(6, 'Audrey Tautou');
```

```
INSERT INTO Direct  
VALUES(6, 'Ron Howard');
```

The values supplied
corresponding to the attributes
in defining the table.

NULL represents "Studio
unknown".

INSERT ...

- ◆ The order for insertion of rows to tables must follow the schema definition:

```
CREATE TABLE Movie(  
    mvID    INTEGER,  
    ....  
    PRIMARY KEY (mvID)  
);  
CREATE TABLE Classification(  
    mvID    INTEGER,  
    ...  
    FOREIGN KEY (mvID) REFERENCES Movie(mvID)  
);
```

INSERT ...

- ◆ So the primary key-rows must be before the foreing key-rows:
 - ◆ INSERT INTO Movies ... (PK mvID=6)
 - ◆ INSERT INTO Classification ... (FK mvID=6)
 - ◆ INSERT INTO Cast ... (FK mvID=6)
 - ◆ INSERT INTO Direct ... (FK mvID=6)

UPDATE

- ◆ Update information for some movies
 - ◆ mvID=5: Change title to "Ice Age 3".
 - ◆ mvID=3: Change title to "Harry Potter 6".

```
UPDATE Movie
SET title = 'Ice Age 3'
WHERE mvID=5;
UPDATE Movie
SET title='Harry Potter 6'
WHERE mvID=3;
```

MVID	TITLE	RA	LEN	STUDIO
3	Happy Potter 6	M	153	Roadshow
5	Ice Age 3	PG	94	20th Century Fox

DELETE

- ◆ The movie with mvID=4 ("The Proposal") is taken off the screening plan and is deleted from the database.
 - ◆ Delete the record with mvID=4 is prohibited --- foreign key constraint violated!

```
SQL> delete from movie where mvID=4;  
delete from movie where mvID=4  
*
```

ERROR at line 1:

ORA-02292: integrity constraint (ZHANG.SYS_C00123875)
violated - child record found

DELETE ...

- ◆ The order for deletion of rows must follow the foreign key constraint.

```
SQL> delete from classification where mvID=4;  
1 row deleted.
```

```
SQL> DELETE FROM Cast where mvID=4;  
5 rows deleted.
```

```
SQL> DELETE FROM Direct WHERE mvID=4;  
1 row deleted.
```

```
SQL> DELETE FROM Movie WHERE mvID=4;  
1 row deleted.
```

DROP

- ◆ Tables, views (discussed later), columns and constraints can be dropped.

`DROP TABLE <table-name>;`

`DROP VIEW <column-name>;`

`ALTER TABLE <tbl-name> DROP COLUMN <col-name>`

`ALTER TABLE <tbl-name> DROP PRIMARY KEY <name>`

- ◆ DROP is different DELETE.

- ◆ DROP data and definition.
- ◆ DELETE data only.

- ◆ ALTER TABLE is different from UPDATE.

- ◆ ALTER TABLE modifies schema.
- ◆ UPDATE modifies data.

The Updated Cinema Database

Movie.

MVID TITLE	RA	Rel_Date	LENGTH STUDIO
-----	----	-----	-----
1 Angels and Demons	M	14-05-2009	138 Sony Pictures
2 Coco Avant Chanel	PG	25-06-2009	108 Roadshow
3 Harry Potter 6	M	15-07-2009	153 Roadshow
5 Ice Age 3	PG	01-07-2009	94 20th Century Fox
6 The Da Vinci Code	M	18-05-2006	

Classification

MVID GENRE

1 Drama
2 Drama
3 Action
3 Adventure
3 Drama
5 Animated
5 Comedy
6 Action
6 Mystery

Cast

MVID ACTOR

1 Tom Hanks
2 Alessandro Nivola
2 Audrey Tautou
2 Benolt Poelvoorde
2 Marie Gillain
3 Daniel Radcliffe
3 Emma Watson
3 Rupert Grint
5 Chris Wedge
5 Denis Leary
5 John Leguizamo
5 Queen Latifah
5 Ray Ramono
6 Audrey Tautou
6 Tom Hanks

SQL3

Direct

MVID DIRECTOR

1 Ron Howard
2 Anne Fontaine
3 David Yates
5 Carlos Saldanha
5 Mike Thurmeier
6 Ron Howard

Views

- ◆ A view is a virtual table defined by an SQL query.
- ◆ Views do not keep data, but can be queried. When a view is queried, it is replaced by its definition to execute the query.

- ◆ View definition:

```
CREATE VIEW <name>AS <view-definition>
```

Views ...

- ◆ With the updated Village Cinema database, a view for movies produced in Roadshow is defined.
- ◆ The view can be queried.

CREATE VIEW RSview1 AS

```
SELECT mvID, title, rating, to_char(rel_date, 'dd-mm-yyyy') as rel_date  
FROM movie  
WHERE studio='Roadshow';
```

```
select *  
from RSview1;
```

Date in the format of dd-mm-yyyy.

The view is queried as a relation.

MVID	TITLE	RA	REL_DATE
2	Coco Avant Chanel	PG	25-06-2009
3	Harry Potter 6	M	15-07-2009

Views ...

- ◆ When defining a view, attributes can be re-named.

```
CREATE VIEW RSview2(mvID, title, cast) AS
```

```
SELECT movie.mvID, title, actor
```

```
FROM Movie, Cast
```

```
WHERE studio = 'Roadshow'
```

```
and Movie.mvID = Cast.mvID
```

```
select *  
from RSview2;
```



MVID	TITLE	CAST
2	Coco Avant Chanel	Alessandro Nivola
2	Coco Avant Chanel	Audrey Tautou
2	Coco Avant Chanel	Benolt Poelvoorde
2	Coco Avant Chanel	Marie Gillain
3	Harry Potter 6	Daniel Radcliffe
3	Harry Potter 6	Emma Watson
3	Harry Potter 6	Rupert Grint

Views ...

- ◆ As views do not keep data, data manipulation operations can not be applied to views. (Special materialized views are exceptions.)

```
SQL> insert into RView2 values(7, 'test', null);  
insert into RView2 values(7, 'test', null)  
*
```

```
ERROR at line 1:  
ORA-01779: cannot modify a column which maps to a non key-  
preserved table
```

NULL Revisited

- ◆ NULL means unknown.
- ◆ The only expression involving NULL that can produce a known value (TRUE or FALSE) is ... IS NULL or ... IS NOT NULL.
- ◆ NULL should be avoided in databases.
 - ◆ Brings ambiguity into databases --- not applicable or not recorded?
 - ◆ Extra care is needed to look after NULL in queries.

NULL Revisited ...

The updated Movie table

MVID	TITLE	RA	Rel_Date	LENGTH	STUDIO
1	Angels and Demons	M	14-05-2009	138	Sony Pictures
2	Coco Avant Chanel	PG	25-06-2009	108	Roadshow
3	Harry Potter 6	M	15-07-2009	153	Roadshow
5	Ice Age 3	PG	01-07-2009	94	20th Century Fox
6	The Da Vinci Code	M	18-05-2006		

```
select mvID, title  
from movie  
where studio is NULL;
```



```
MVID TITLE  
-----  
6 The Da Vinci Code
```

Checking if studio is NULL. Result in TRUE or FALSE

```
select mvID, title  
from movie  
where studio=NULL;
```



```
no rows selected
```

Comparing the value of studio with NULL, resulting in UNKNOWN.

Comparing NULL to Values

- ◆ Comparing any value (including NULL itself) with NULL yields UNKNOWN.
- ◆ In the WHERE clause of a query, a tuple is chosen iff the WHERE clause is TRUE— **not FALSE or UNKNOWN**.

NULL and Aggregation

- ◆ NULL is not a known value, and so it is not counted when doing aggregation.

The updated Movie table

MVID TITLE	RA	Rel_Date	LENGTH STUDIO
-----	---	-----	-----
1 Angels and Demons	M	14-05-2009	138 Sony Pictures
2 Coco Avant Chanel	PG	25-06-2009	108 Roadshow
3 Harry Potter 6	M	15-07-2009	153 Roadshow
5 Ice Age 3	PG	01-07-2009	94 20th Century Fox
6 The Da Vinci Code		18-05-2006	

NULL And Aggregation ...

- ◆ Do the following queries return the same result?

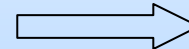
```
select studio, count(*)  
from movie  
group by studio
```



STUDIO	COUNT(*)

	1
Sony Pictures	1
Roadshow	2
20th Century Fox	1

```
select studio, count(rating)  
from movie  
group by studio
```



STUDIO	COUNT(*)

	1
Sony Pictures	1
Roadshow	2
20th Century Fox	1

```
select studio, count(studio)  
from movie  
group by studio
```



STUDIO	COUNT(*)

	0
Sony Pictures	1
Roadshow	2
20th Century Fox	1